IN THE CLAIMS

Please amend claims as follows:

1. (Previously Presented) A method, comprising the steps of:

loading a code module into a memory space of a first domain, wherein the first domain owns one of a kernel space and a portion of a user space, the code module including an instruction having a symbol reference;

determining if the symbol reference is to an external location outside of the memory space;

generating a link stub for the symbol reference when the symbol reference is to an external location to access the external location;

redirecting the instruction to the link stub; and

determining if the external location is within a second domain that is within a protection view of the first domain, wherein the second domain owns the other one of the kernel space and a portion of the user space;

requesting attachment of the second domain to the first domain when the second domain is determined not to be within the protection view of the first domain; and

attaching the second domain to the first domain using an attachment mechanism.

2. (Original) The method of claim 1, wherein the link stub is part of a linking table entry corresponding to the symbol reference.

3. (Original) The method of claim 1, wherein the link stub is a jump instruction to the external location.

4. Cancelled

5. (Previously Presented) The method of claim 1, further comprising the steps of:

determining whether the attachment request is permitted based on authorization

information provided by the first domain;

wherein attachment to the second domain is not permitted when the attachment request is not permitted.

6. Cancelled

7. Cancelled

8. (Previously Presented) A method, comprising:

creating a task in a first domain, wherein the first domain owns one of a kernel space and a portion of a user space, the task executing a number of instructions;

executing a first jump instruction in the number of instructions that refers to a link stub corresponding to an external location in a second domain, wherein the second domain owns the other one of the kernel space and a portion of the user space;

executing the link stub;

comparing the external location to a task protection view;

generating a processing exception when the external location is outside the task protection view; and

executing an exception handling routine in response to the generation of the processing exception, the exception handling routine including,

saving a pre-exception setting of the task protection view,

altering the task protection view to include a protection view of the second domain, and

jumping to the external location.

9. (Original) The method of claim 8, wherein the link stub is part of linking table entry corresponding to the external location.

10. (Previously Presented) The method of claim 8, wherein the link stub includes a second

jump instruction to the external location.

11. Cancelled

12. Cancelled

13. (Previously Presented) The method of claim 8, wherein the task protection view is saved on a task protection switch stack.

14. (Previously Presented) The method of claim 8, further comprising steps of:

retrieving the pre-exception setting of the task protection view;

restoring the task protection view using the pre-exception setting of the task protection view;

returning to a subsequent instruction to the first jump instruction in the number of instructions.

15–34. Cancelled